

## Python Programmierung 2

13.11.2023 bis 22.01.2024

Durchführung: 23H\_\_PYT2\_22NI

Lehrperson: Walter Rothlin

Block	Seq	Dauer	Art	Inhalt	Ziele	Methode	Hausaufgabe
01	1	4.00L	PU	<ul style="list-style-type: none"> <li>* Repetition class(), methoden, instance-variablen, properties</li> <li>* private / public</li> <li>* Statische Methoden</li> <li>* docStrings</li> <li>* Vererbung</li> <li>* Reflaction (Zugriff auf docStrings, dynamisch Function-Call)</li> <li>* Enumerations (ChatJPT: Good morning. Pls show me some examples for enums in Python)</li> <li>* Sub-Classes (e.g. Sense\_Hat)</li> </ul>	Vertiefen des Klassen-Konzept in Python	Problem-Based	Code-Refactoring
02	1	4.00L	PU	<ul style="list-style-type: none"> <li>* Requirements überarbeiten</li> <li>* Interface (Abwärtskompatible) überarbeite (Cleancode)</li> <li>* Public/private/Properties/setter/getter</li> <li>* Naming</li> <li>* ENUM für Log-Level</li> <li>* docStrings / API Doc</li> </ul> Refactoring Weather Application (Usage of Logger-Klasse)	Refactoring Logger-Klasse	Problem-Based	Code-Refactoring
03	1	1.20L	PU	Leistungskontrolle 1 (Design and Implement a Class)			
03	2	2.80L	PU	<ul style="list-style-type: none"> <li>* Unterschied Packages / Module</li> <li>* Dokumentation von Modulen</li> <li>* Namespaces</li> <li>* Selektiver Import</li> <li>* Abfangen von Exceptions: Try / except / else / finally</li> <li>* Exception werfen: Raise</li> <li>* Exception-Handling: lösen/werfen</li> <li>* Eigene Exceptions</li> </ul>	<ul style="list-style-type: none"> <li>* Umgang mit Modulen und Packages</li> <li>* Exception Handling in Python</li> </ul>	Problem-Based	Code-Refactoring
04	1	4.00L	PU	<ul style="list-style-type: none"> <li>* Threads kreieren (fork) / stoppen</li> <li>* Kommunikation zwischen Threads</li> <li>* Thread-Synchronisation</li> <li>* Locks</li> <li>* I/O Interrupts</li> <li>* Time-Interrupts</li> </ul>	Multithreading in Python	Problem-Based	Code-Refactoring
05	1	1.20L	PU	Leistungskontrolle 2 (Design and Implement a Sub-Class)			
05	2	2.80L	PU	<ul style="list-style-type: none"> <li>* GPIO (LED-Steuerung)</li> <li>* PiPlates</li> <li>* Relais-Karte</li> <li>* Analog/digital I/O</li> </ul> Mutteruhr für SBB-Uhr Selecta-Automat	Hardware ansteuern	Problem-Based	Code-Refactoring
06	1	4.00L	PU	<ul style="list-style-type: none"> <li>* Template-Mechanism (Jinja2)</li> <li>* ChatJPT: Give me an example for JINJA Templates?</li> </ul>	Template Mechanismus	Problem-Based	Code-Refactoring



Block	Seq	Dauer	Art	Inhalt	Ziele	Methode	Hausaufgabe